

# Text to Speech

Note: This feature is only available on the following devices: Roku Streaming Stick (3600X), Roku Express (3700X) and Express+ (3710X), Roku Premiere (4620X) and Premiere+ (4630X), Roku Ultra (4640X), and any Roku TV running Roku OS version 7.2 and later.

## Table of Contents

- Text to Speech Components
- Audio Guide Behavior for SceneGraph Nodes
- Audio Guide Support for BrightScript Components
- Implementation Tips
  - TTS Interruptions
  - Other TTS Implementation Changes
  - Long Text Delays

## Text to Speech Components

*Components available since firmware version 7.2*

Text to speech (TTS) allows you to provide an audible spoken version of the strings shown to the user in your application. For platforms that are required to comply with the [FCC Communications and Video Accessibility Act of 2010 \(CVAA\)](#), this capability can be used as part of compliance with CVAA, and the current text to speech flite\_tts library is built into the image. The Roku text to speech capability supports different languages, voices, rates of speech, volume of speech, and other aspects of text to speech. Roku provides text to speech support in the following components, interfaces, and events:

- `roTextToSpeech`
- `ifTextToSpeech`
- `roTextToSpeechEvent`

*Components available since firmware version 7.5*

- `roAudioGuide`
- `ifAudioGuide`

## Audio Guide Behavior for SceneGraph Nodes

- **ArrayGrid**: speaks focused item (`ContentMetaData::TITLE`), followed by navigation hint, then `ContentMetaData::AUDIO_GUIDE_SUFFIX` (if any).
- **Button**: text of button is spoken only if focused
- **ButtonGroup**: speaks focused `Button`, followed by navigation hint ("button 1 of 4"), followed by button-specific hint, if any. (Button-specific hint is spoken only for `StarRatingButton`.)
- **CheckList**: speaks focused item (`ContentMetaData::AUDIO_GUIDE_TEXT` if any; otherwise `ContentMetaData::TITLE`) followed by navigation hint ("checkbox, checked, 1 of 4")
- **Dialog**: speaks `title`, `message`, and `bulletText` (if any), then reads focused button
- **Keyboard**: speaks hint about caps lock toggling (once), then speaks focused key
- **KeyboardDialog**: speaks `title`, then keyboard

- **Label:** speaks `text` field
- **LabelList:** speaks focused `ContentMetaData::AUDIO_GUIDE_TEXT` if any; otherwise speaks `ContentMetaData::TITLE`, followed by navigation hint.
- **MarkupGrid:** speaks focused `ContentMetaData::AUDIO_GUIDE_TEXT` if any; otherwise speaks `ContentMetaData::TITLE`, followed by navigation hint, then `ContentMetaData::AUDIO_GUIDE_SUFFIX` (if any), then MEDIA speech (see below)
- **MarkupList:** speaks focused item (`ContentMetaData::TITLE`), followed by navigation hint, then `ContentMetaData::AUDIO_GUIDE_SUFFIX` (if any).
- **MiniKeyboard:** speaks focused key
- **PinDialog:** speaks dialog title, whether in key pad, then focused key or button
- **PinPad:** speaks focused key
- **Poster:** if focused, speaks `audioGuideText` field (if set)
- **PosterGrid:** speaks focused item (`ContentMetaData::TITLE`), followed by navigation hint.
- **ProgressDialog:** speaks dialog `title`, `message`, and `bulletText` every 15 seconds. Speaks focused button if there is any.
- **RadioButtonList:** speaks focused item (`ContentMetaData::AUDIO_GUIDE_TEXT` if any; otherwise, `ContentMetaData::TITLE`), followed by navigation and selection hint
- **RenderableNode:** if speaking focused item (depends on context), will speak focused descendant; otherwise, will speak all descendants
- **RowList:** speaks row label (when row becomes focused), then speaks focused **PosterGrid** or **MarkupGrid** (MarkupGrid is used if `itemComponentName` is non-empty)
- **ScrollableText:** speaks `text` field
- **ScrollingLabel:** speaks `text` field
- **Video:** speaks HUD if displayed by user

#### Audio Guide behavior for built-in SceneGraph panels and scenes:

- **GridPanel:** speaks panel, then **leftLabel**
- **ListPanel:** speaks panel, then **leftLabel**
- **PanelSet:**
  - If left panel is focused, speaks focused left panel, then unfocused right panel (if any)
  - If right panel is focused, speaks unfocused left panel, then focused right panel
  - If no panel is focused, speaks unfocused left panel, then unfocused right panel (if any)
- **OverhangPanelSetScene:** uses **Overhang** title when speaking location
- **Scene:** speaks dialog (if any); otherwise speaks **PanelSet** (if any); otherwise speaks as **RenderableNode**

#### MEDIA speech is spoken in the following order:

- `ContentMetaData::TEXT`
- `ContentMetaData::DESCRIPTION`
- `ContentMetaData::DIRECTORS`
- `ContentMetaData::PRODUCERS`
- `ContentMetaData::ACTORS`

#### There is no additional speech for the following nodes (they will behave the same as **RenderableNode**):

- `BifDisplay`
- `BusySpinner`

- LayoutGroup
- Overhang
- Panel
- ProgressBar
- Rectangle
- TextBox
- TrickPlayBar

## Audio Guide Support for BrightScript Components

- roPosterScreen
- roGridScreen
- roMessageDialog
- roParagraphScreen
- roListScreen
- roKeyboardScreen
- roSpringboardScreen

## Implementation Tips

### TTS Interruptions

Many channel UI elements have default TTS behavior. It is possible that speech triggered by these implementations can interrupt your TTS implementation at times. You should keep track of the IDs of your TTS utterances, as returned by `say()` and `silence()`, and handle interruptions accordingly.

### Other TTS Implementation Changes

Other TTS implementations may change the current voice, the current language, the current volume, the current pitch, and/or the current speech rate. You should keep track of how these parameters might change.

### Long Text Delays

A long text string to be spoken by TTS may have a noticeable delay before starting the speech, at least for the first speech of the long string. For long text strings, you can break up the text string so that the first speech is a reasonably short sentence, followed by longer sentences as needed. You should not break up the long text string into individual words, as it will affect phrasing without improving the perceived delay in any noticeable way.